(c) associating the user with roles;

(d) creating a user context instance upon successful identification of the user, wherein the user context instance includes information about the user including the roles;

(e) receiving a request from the user to invoke a first service on a first component, wherein the first component invokes a second service of a second component, and wherein completion of the second service is necessary to complete the first service;

(f) querying the user context for the information about the user;

(g) comparing the user information with an access control list for verifying that the user has access to the first component; and

(h) comparing the user information with an access control list for verifying that the user has access to the second service of the second component.

5. A method as recited in claim 4, wherein the first service invoked associates any objects created, updated, or deleted as a result of the invocation of the first service with the user context instance.

7. A computer program embodied on a computer readable medium for maintaining a security profile throughout nested service invocations on a distributed, component-based system, comprising:

(a) a code segment that provides interconnections between distributed components each having nested service invocations;

(b) a code segment that identifies a user;

(c) a code segment that associates the user with roles;

(d) a code segment that creates a user context instance upon successful identification of the user, wherein the user context instance includes information about the user including the roles;

(e) a code segment that receives a request from the user to invoke a first service on a first component, wherein the first component invokes a second service of a second component, and wherein completion of the second service is necessary to complete the first service;

(f) a code segment that queries the user context for the information about the user;

(g) a code segment that compares the user information with an access control list for verifying that the user has access to the first component; and

(h) a code segment that compares the user information with an access control list for verifying that the user has access to the second service of the second component.

11. A computer program as recited in claim 10, wherein the first service invoked associates any objects created, updated, or deleted as a result of the invocation of the first service with the user context instance.

13. A system for maintaining a security profile throughout nested service invocations on a distributed, component-based system, comprising:

(a) logic that provides interconnections between distributed components each having nested service invocations;

(b) logic that identifies a user;

(c) logic that associates the user with roles;

(d) logic that creates a user context instance upon successful identification of the user, wherein the user context instance includes information about the user including the roles;

(e) logic that receives a request from the user to invoke a first service on a first component, wherein the first component invokes a second service of a second component, and wherein completion of the second service is necessary to complete the first service;

(f) logic that queries the user context for the information about the user;

(g) logic that compares the user information with an access control list for verifying that the user has access to the first component; and

(h) logic that compares the user information with an access control list for verifying that the user has access to the second service of the second component.

17. A system as recited in claim 16, wherein the first service invoked associates any objects created, updated, or deleted as a result of the invocation of the first service with the user context instance.